

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 02-195462

(43)Date of publication of application : 02.08.1990

(51)Int.Cl.

G06F 13/10
G06F 9/46

(21)Application number : 01-015347

(71)Applicant : MATSUSHITA ELECTRIC IND CO LTD

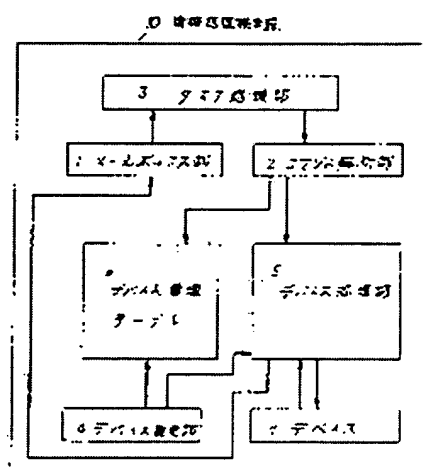
(22)Date of filing : 24.01.1989

(72)Inventor : FUJIMOTO KAZUO

(54) INFORMATION PROCESSING TERMINAL EQUIPMENT

(57)Abstract:

PURPOSE: To make it possible to change and add an I/O device and to make the correspondence of the change and addition flexible by allowing a command analyzing part to refer the contents of a device control table at the time of using the device. **CONSTITUTION:** At the time of generating an I/O command after specifying a device and setting up a device parameter from the side of a task processing part 3, the command analyzing part 2 receives the command and retrieves whether the device concerned exists in the device control table 6 or not. When the specified device exists and the device is not acquired yet, a device processing part 5 is started and device operation is started. After ending the event of the device 7, a device processing part 5 returns a mail message to a mail box part 1.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

⑩ 日本国特許庁(JP)

⑪ 特許出願公開

⑫ 公開特許公報(A) 平2-195462

⑬ Int.Cl.⁵

識別記号

庁内整理番号

⑭ 公開 平成2年(1990)8月2日

G 06 F 13/10
9/46

3 2 0 A
3 4 0 B

7737-5B
8945-5B

審査請求 未請求 請求項の数 8 (全12頁)

⑮ 発明の名称 情報処理端末器

⑯ 特 願 平1-15347

⑰ 出 願 平1(1989)1月24日

⑱ 発 明 者 藤 本 和 生 大阪府門真市大字門真1006番地 松下電器産業株式会社内
⑲ 出 願 人 松下電器産業株式会社 大阪府門真市大字門真1006番地
⑳ 代 理 人 弁理士 栗野 重孝 外1名

明 細 書

1、発明の名称

情報処理端末器

2、特許請求の範囲

(1) 端末器内のデバイス情報を保有するデバイス管理テーブルと、デバイス入出力処理を行うデバイス処理部と、前記デバイス処理部の実行結果を格納し、要求される場所に結果を送る機能を有するメールボックス部と、前記デバイス処理部のデバイス処理に必要なコマンドを解析する機能を有するコマンド解析部と、前記コマンド解析部に必要なコマンドを発行することにより、各種の情報処理を実現するタスク処理部と、端末器内のデバイス情報を変更する機能を有するデバイス変更部とを備え、前記デバイス変更部は、前記デバイス処理部の再登録と同時に前記デバイス管理テーブルの内容を書き換えることによって、デバイス情報の変更を可能にする手段を有する情報処理端末器。

(2) デバイス管理テーブルに新規デバイス情報を

書き込み、デバイス処理部を追加登録する手段を有する請求項1記載の情報処理端末器。

(3) コマンド解析部が、デバイス管理テーブルを検索し、保有するデバイスの有無を判定する機能を有する請求項2記載の情報処理端末器。

(4) デバイス管理テーブル内のデバイス情報登録時に、保有するデバイスのうちで、高速処理が必要とされるデバイスを優先的に扱い、デバイス情報を再配置する手段を有する請求項2記載の情報処理端末器。

(5) デバイス処理部を、直接デバイスハードウェアの制御を行う機能を有するデバイスドライバ部と、デバイスハードウェアのみでは実現不可能な機能を拡張し、かつデバイス処理結果をメールボックス部に通知する機能を有するデバイスハンドラ部で構成した請求項1または請求項2記載の情報処理端末器。

(6) 一つのデバイスに対し、一つのデバイスドライバ部と、一つ以上のデバイスハンドラ部で構成することで、一つのデバイスを用いて、複数

の使用目的を実現する手段を有する請求項 6 記載の情報処理端末器。

(f) デバイス管理テーブル内に、一つのデバイスドライバ部と、複数のデバイスハンドラ部の内で特定のデバイスハンドラを指定する連結情報を格納する手段を有する請求項 6 記載の情報処理端末器。

(g) デバイス管理テーブル内に、デバイスの使用状態を示す情報を格納するデバイスステータスを有し、コマンド解析部が、コマンド解析時に前記デバイスステータスを見ることで、マルチタスクオペレーティングシステム下で二つ以上のタスク処理部が、同時に同一デバイスの使用を禁止する競合状態の排除機能を実現する請求項 1 記載の情報処理端末器。

3. 発明の詳細な説明

産業上の利用分野

本発明は複数の入出力デバイスを備え、同時に複数のデバイスを動作させて、ユーザにサービスを提供する情報処理端末器に関するものである。

用し、サービス利用者に、どのようなサービスの適用を受けたいかを選ばせるために、表示した何種類かのサービスのキーボード入力待ちの状態と想定する。まずタスク処理部 33 にて、使用したいデバイス 37 の使用要求をコマンドとして発行する。第 10 図では、コマンド解析部 32 とデバイス処理部 35 を、まとめてデバイス処理 42 として示している。コマンドには利用パラメータ（指定デバイス名、入力文字数等）の情報を含んでいるものとする。コマンド解析部 32 では前記パラメータの有効性を判定し、有効であれば、デバイス処理部 35 へ制御を移し、無効であれば、タスク処理部 33 にパラメータ無効の結果を戻す。パラメータが有効の時には、デバイス処理部 35 にて該当するデバイス 37 を起動する。今キーボード用の割り込み入力があれば、キーが入力されたことが判定できるように構成されていたとすると、デバイス処理部 35 はここで割り込み入力待ち状態になる。この時デバイス 37 の使用依頼を出したタスク処理部 33 内のタスクプログラムも、

従来の技術

近年はサービス業務の多様化により、携帯用や、据置型の様々な情報処理端末器が存在する。これら端末器は内蔵されているか、もしくは外部に接続される複数のデバイスを制御している。従来の情報処理端末器は、各デバイス制御を実現するために、第 9 図に示すような構成をとっていた。第 9 図において、40 はここで説明する情報処理端末器、37 は情報処理端末器 40 内のデバイス、33 は前記ハードウェアデバイスを動作させるために有効なデバイス 37 を選択しながら、各種のアプリケーションプログラムを実行するタスク処理部、32 は前記タスク処理部 33 から発せられたデバイス 37 用のコマンドを解釈し、選択されたデバイス 37 に起動をかける機能を有するコマンド解析部、35 は実際にデバイス 37 を動作させる機能を有するデバイス処理部である。

以下第 10 図に従って動作を説明する。情報処理端末器 40 内のデバイス 37 を、キーボードと設定して説明する。この情報処理端末器 40 を選

該デバイス処理部 35 からの処理結果待ちで止まっている。

タスクプログラムはデバイス 37 の待ち状態が発生すると、待ち状態以外の別の処理が行えない。そこでマルチタスクオペレーティングシステムを導入し、あるタスクプログラムが待ち状態に入った時には、次のタスクプログラムが別の処理を行うように構成することで、この待ち状態の時に、他の処理を行うことができる。勿論待ち状態が回避されたときには、先のタスクプログラムの処理を続行する。

キーボードの割り込み入力が発生すると、デバイス処理部 35 がデバイス処理を開始する。処理が終了すると、キーボードの割り込み入力を禁止して、処理結果（例えば、キーボード入力番号）をタスク処理部 33 に通知する。タスク処理部 33 は、処理結果を受け付けると、選択されたサービスの実行を開始する。

情報処理端末器 40 は、複数のデバイス 37 を保有しており、そのデバイス 37 を起動するため

のコマンド及び、デバイス処理部35をもっている。これらのデバイス処理部35は、一般にデバイスドライバとも言われるが、オペレーティングシステムに、このデバイスドライバの機能を組み込んだ、入出力ドライバー体型のオペレーティングシステムを構築している例が多い。情報処理端末器40のプログラムの開発においては、製造コストの削減と、納期短縮のためデバイスハードウェアを限定し、開発プログラム量の削減をはかるためである。そのため通常は、タスク処理部33と、デバイス処理部35を同時に開発し、一緒にアセンブルもしくはコンパイルし、開発機側でリンク作業と呼ばれるプログラムの結合作業を行って、リンク後のオブジェクトプログラムを情報処理端末器40のROM等の不揮発性で書き換え不可のメモリにおさめる。

発明が解決しようとする課題

しかしながら前記のような情報処理端末器の構成では、次のような課題があった。

第1に入出力ドライバー体型のオペレーティン

る機能を有するコマンド解析部と、前記コマンド解析部に必要なコマンドを発行することにより、各種の情報処理を実現するタスク処理部と、端末器内のデバイス情報を変更する機能を有するデバイス変更部とを備え、前記デバイス変更部は、前記デバイス処理部の再登録と同時に前記デバイス管理テーブルの内容を書き換えることによって、デバイス情報の変更を可能にする手段を有するものとしたものである。

作用

この構成により、デバイス処理部内のデバイス存在情報は、全てデバイス管理テーブルに記述されており、デバイス使用開始時に、コマンド解析部がそのテーブルの内容を参照することによって、デバイス処理を起動することが可能であり、またデバイス管理テーブルの内容を変更、追加できる機能を持つことで、多様なシステム要求に柔軟に対応することができる。

実施例

以下本発明の実施例を説明する。第1図は本発

明の第1の実施例の情報処理端末器の構成図である。第1図において、10はここで説明する情報処理端末器、7は情報処理端末器10内の入出力装置であるデバイス、5はデバイス7を直接制御するデバイス処理部、6は情報処理端末器10内のデバイス7の保有情報等を格納するデバイス管理テーブル、2は指定デバイス7の選択を行うコマンド解析部、4はデバイス処理内容の変更を行うデバイス変更部、1はデバイス処理部5の処理結果を格納するメールボックス部、3は入出力コマンドを発行したり、デバイス処理以外の演算処理を行う機能を有するタスク処理部である。

この情報処理端末器10の内部の処理の手順を以下に示す。ここでデバイス7に、キーボード入力装置を想定して説明する。先ずメールボックス部1内に、キーボード入力装置の処理結果を格納するメールボックスを生成する。次に第2図のように、タスク処理部3側からデバイス指定、デバイスパラメータを設定したりえて、入出力コマンドを発行する。デバイス処理12側のコマンド解

析システム構築方式では、デバイス処理部内の各デバイス処理プログラムが容易に変更、追加できる構造を持っていないため、ユーザが新規に開発した入出力デバイス、もしくは購入した入出力デバイスを追加することが困難であった。

第2に想定される入出力デバイスの処理プログラムを予め全て持つ場合は、デバイスの数が多くなると、デバイス処理プログラムのサイズが飛躍的に大きくなる問題があった。

本発明はこのような課題を解決するもので、入出力デバイス処理プログラムを逐次的に容易に追加できる機能をもった情報処理端末器を提供するものである。

課題を解決するための手段

前記課題を解決するために本発明は、端末器内のデバイス情報を保有するデバイス管理テーブルと、デバイス入出力処理を行うデバイス処理部と、前記デバイス処理部の実行結果を格納し、要求される場所に結果を送る機能を有するメールボックス部と、デバイス処理に必要なコマンドを解析す

明の第1の実施例の情報処理端末器の構成図である。第1図において、10はここで説明する情報処理端末器、7は情報処理端末器10内の入出力装置であるデバイス、5はデバイス7を直接制御するデバイス処理部、6は情報処理端末器10内のデバイス7の保有情報等を格納するデバイス管理テーブル、2は指定デバイス7の選択を行うコマンド解析部、4はデバイス処理内容の変更を行うデバイス変更部、1はデバイス処理部5の処理結果を格納するメールボックス部、3は入出力コマンドを発行したり、デバイス処理以外の演算処理を行う機能を有するタスク処理部である。

この情報処理端末器10の内部の処理の手順を以下に示す。ここでデバイス7に、キーボード入力装置を想定して説明する。先ずメールボックス部1内に、キーボード入力装置の処理結果を格納するメールボックスを生成する。次に第2図のように、タスク処理部3側からデバイス指定、デバイスパラメータを設定したりえて、入出力コマンドを発行する。デバイス処理12側のコマンド解

析部2は、このコマンドを受け付け、デバイス管理テーブル6の中に該当デバイス7が存在するかどうかを検索する。ここでデバイス処理12とは、コマンド解析部2とデバイス処理部5とデバイス管理テーブル6を含む。必要な入出力デバイス7を起動するコマンド解析部2は、デバイス7に応じた入出力コマンドを待つ。例えばキーボード入力コマンドには、専用の「KEY IN」コマンドが用いられる。コマンド解析部2はこのコマンドが、どのデバイス7のどの処理に該当するかを判別し、デバイス7の動作パラメータを設定し、処理をデバイス処理部5に移すものである。

このデバイス管理テーブル6の構成例を示した例が第3図である。第3図は、この情報処理端末器10に登録されているデバイス7の名称と、そのデバイス処理部5が存在するアドレスと、デバイス7が現在何かの入力待ち状態であるとか、現在動作中であるというデバイス獲得及び使用状態を示すデバイスステータス情報を格納している。このデバイス管理テーブル6の情報を変更、追加

当するメールメッセージが届くまで待つ。従ってマルチタスクオペレーティングシステム下でも、従来例のようにタスク処理部3内の複数のタスクプログラムを動作させずとも、メールメッセージ待ちを行わないかぎり、連続して次から次と、処理を続行することができる。勿論、複数のタスクプログラムを動作させても良い。

キーボードの割り込み信号を受け付けると、デバイス処理を開始し、処理が終了すれば、デバイス7の割り込み入力を受け付けないようにして、デバイス処理を終了させる。この処理結果は、メールメッセージとしてメールボックス部1に送られる。このデバイス処理部5は、デバイス7の種類に応じて、割り込み入力で動作するものあれば、割り込み入力を伴わないものもある。割り込みを伴わないデバイス処理は、通常は待ち時間が不必要であるが、この割り込みを伴わないデバイス7の処理を実行している最中に、他の割り込み入力があれば、現在の処理を中断して、該当デバイス処理部5が起動されるように構成すると、後者

する方法については、後で詳細に述べる。またデバイス7の状態を示す例についても後で詳しく述べる。

タスク処理部3が指定したデバイス7が存在し、その獲得状態が未獲得状態であれば、この該当するデバイス処理部5を起動し、デバイス7の動作を開始する。ここではキーボード入力装置を動作状態に設定し、ある特定のキーボード入力を割り込み入力信号として、割り込み入力があれば、入力判断処理を開始する。従来例と異なるのは、入力待ち処理に移る前に、タスク処理部3に対して、該当デバイス7が使用許可状態に変わった応答を返す構成をもつ。この構成を待つことでタスク処理部3は、指定デバイス処理以外の演算処理等を開始することが可能である。この間に例えば、本日のキャッシュサービスの集計などの計算処理を行うことも可能である。また他のデバイス7に対して、入出力デバイス要求のコマンドを発行することもできる。デバイス7の処理結果が欲しい時には、メールボックス部1の機能を利用して、該

の割り込みをともなった割り込み処理プログラムが終了するまでの時間は、待たねばならない。またユーザが割り込みを伴うデバイス7と、割り込みを伴わないデバイス7の区別を意識せずともプログラム開発が出来るように、全てのデバイス処理部5の処理の終了はメールボックス部1を経由して通知する構成とする。またメールボックス部1内のメールボックスの個数は、同時に並行して動作するデバイス7の総数以上で構成する必要がある。メールボックス部1の機能は、欲しいメールメッセージが受信できるまで待ち続ける機能以外に、メールメッセージが届いているか否かを見て、届いている時はメールメッセージを受け取り、届いていないときは、何のメールメッセージ処理を行わずに、次の処理を開始するような機能をもつように構成してもよい。前記のようにまだ届いていない時は、別の処理を行うように構成すると、処理の多重性が高まる。

各デバイス7に対応するデバイス処理部5は、該当デバイス7の事象が終了すれば、その結果を

メールメッセージとしてメールボックス部1に返す。正常に処理が終了したということを知らせるだけでなく、該当デバイス7が故障しているとか、デバイス異常が検知されたとか、またはパラメータに不備がみられたとかの内容をメールメッセージとして、メールボックス部1に送信する。

このメールメッセージの内容は、現在のデバイス状態と、終了状態というように複数のメッセージから構成するようにしてもよい。例えば、正常、パリティエラー（データ受信処理中に発見された異常で、正しくデータが受信できなかったことを示す）、タイムアウトエラー（制限時間内に、処理が終了しなかったことを示す）、強制終了という終了状態の詳細な内容を返すように構成することもできる。メッセージはこの例の他に、実際のデータそのものを含むように構成してもよい。例えば、キーボードデバイスのときには、どのキーボードが押されたかを示すキー番号をメールメッセージに含めても良い。また10バイトデータを受信する処理を行っていて、8バイトめまでは正

しく受信できたが、9バイトめでパリティエラーが発生したときは、前記8バイトのデータ受信とパリティエラー発生の2種の内容をメッセージに含めるように構成してもよい。

デバイス7は例えば、必要な時間毎に割り込み入力が発生するリアルタイムクロックであるとか、キーボード等の入力が発生したら割り込みが発生する入力デバイスや、他の機器とデータ通信を行う際に、入力があれば割り込みが発生し、もしくはデータを送信できるように準備が整えば、割り込みが発生するような通信用デバイスをハードウェアで構成する。その他割り込みを伴わずとも動作可能なスピーカデバイス等のデバイス7も存在する。

割り込みを伴う各種のデバイス7に対して、パラメータで設定された条件が満たされる回数だけ割り込みを待つ構成とする。例えば3バイトのデータを受信するパラメータを設定し、1バイトの受信データがあれば割り込みが発生するようにデバイス7を構成すれば、3回受信割り込みを待つ

て、データの受信を完了するまで、該当デバイス7の割り込みを許可し、3回あった時点で、割り込みを禁止し、該当デバイス7を不活性化するように構成する。デバイス7は、このように一つの処理しか行わないものもあるし、デバイス7そのものがインテリジェント化されていて、マイクロコンピュータ等を実装し、あるコマンドを送ればデバイス7側で処理が行われて、デバイス処理終了後、結果がレスポンスとして受信できるように構成する場合もある。

デバイス処理内容を変更する時には、デバイス変更部4を起動し、変更したいデバイス処理部5をロードする。この時デバイス変更部4は、デバイス管理テーブル6を参照し、メモリの空き領域を調べ、そこにデバイス処理部5の該当プログラムをロードする。デバイス処理プログラムは、情報処理端末器10のメモリのうち、どこにでも適当な位置に配置しても構わないように、プログラムコードが相対番地で表されるリロケータブル形式で記述されている必要がある。リロケータブル

形式で記述されていない場合は、デバイス管理テーブル6の内容を読みだして、空いているメモリを捜し出し、その絶対番地でプログラムする必要がある。

デバイス管理テーブル6の変更方法は、幾種類か想定される。

例えば第3図のように、デバイス管理テーブル6に次から次に追加していく方法である。第3図において、デバイス名AからKは端末器に初めからプログラムされているデバイス名及びデバイスプログラムアドレスである。デバイス名Cの処理内容を変更したい場合、変更領域にデバイス名Cと、デバイスプログラムアドレスを書き込む構成とする。変更領域の内容を優先して検索する方法をとれば、該当デバイス7の処理内容を変更することができる。第4図はデバイス管理テーブル6のメモリ構成を示したものである。初めから端末器に固有の情報を固定情報部61に、変更した情報を変更情報部62に格納する。そして変更情報部62を優先的に検索する機能を実現するのがデ

バイス選択部60である。固定情報部61は不揮発性の書き換え不可能なメモリで構成し、変更情報部は不揮発性の書き換え可能なメモリ(EEPROMやバッテリーバックアップされたRAM)で構成する。デバイス選択部60は、デバイス管理テーブル6のメモリ構成等の位置情報と、優先権情報(ここでは変更情報部62を優先するという規約)を利用して、デバイス情報を検索し、動作を決定する権限を持つ。

またデバイス管理テーブル6をすべて不揮発性の書き換え可能なメモリで構成し、変更情報は該当するデバイス情報に上書きをする構成としても良い。この時にも、デバイス処理部5はメモリの空き領域を調べて、その領域を割り当てるように構成する。割り当てる領域がなくなった時には、どれか不要なデバイス処理プログラムを削除できる手段を持てば、さらに柔軟性が高まる。

デバイス管理テーブル6の内容の変更は、ここで述べているデバイス変更部4のような、専用のテーブル変更処理部を持つ場合と、ユーザが直に

この内容を書き換える場合が存在する。しかしユーザが勝手に書き直して、正常に動作しなくなった場合は、解析が困難になるので、これを許さない方法をとる方が良い。テーブルの管理は、デバイス変更部4等のシステムソフトウェアに委ねることが好ましい。またセキュリティ保護の点からこのデバイス管理テーブル6はユーザには解放しない方が良くと思われる。なぜならばデバイスに関する情報を勝手に書き換えることで、不正使用が可能になる危険性があるからである。

第2の実施例として、デバイス処理内容の変更だけでなく、デバイス7及びデバイス処理部5の追加が可能な情報処理端末器10について述べる。情報処理端末器10の構成要素及び、使用方法是第1の実施例と同様である。ただしデバイス変更部4は、デバイス処理内容の変更だけでなく、追加の要求も受け付ける。追加内容はデバイス管理テーブル6内の変更情報部62に書き込む構成とする。追加のデバイス名が、現在所有するデバイス名と同一とならないように配慮する必要がある。

従ってデバイス変更部4は、デバイス登録時に同じ名称になれば、これを排除し、再度別の名称を要求する機能が必要である。もしくは内部で自動的に別の名称を付け、変更された名称を返すように構成してもよい。

第3の実施例は、デバイス処理要求を判別し、指定デバイス処理部5を起動するコマンド解析部2の他の構成例を示す。コマンド解析部2は、まずデバイス管理テーブル6を検索し、該当デバイス7が存在するかどうかを判定し、該当デバイス7が見つければ、そのデバイス処理部5を直接起動するまでの役割のみを分担する構成とする。このように構成することで、デバイス固有の発行コマンドを持たずとも、該当のデバイス処理を起動できる。第1のコマンド解析部2が行っていたパラメータ処理は、そのあとすぐに起動されるデバイス処理部5内で行えばよい。

コマンド解析の方法には下記のコマンド処理用のシステムコールと呼ばれる方法を用いる。これはコマンドを実行するに最低限必要なパラメータ

をレジスタや、メモリに格納した後で、ソフトウェア割り込みを利用して、マイクロプロセッサの割り込みポイントテーブル領域に制御を移すものである。例えば、「INT n」(インテル社8086の場合)命令を実行すると、該当する割り込みポイントテーブル領域に記述してある割り込みサービスアドレスにジャンプする。この命令を利用することで、端末器外部で作られたアプリケーションプログラムは、内部のデバイスのデバイス処理プログラムのアドレスを知らずとも、デバイス名を示すデータを設定した後で、このソフトウェア割り込みを発生するプログラムを書くだけで、デバイス処理を実現できる。

デバイス処理をシステムコールという方法を使わず、タスク処理部3内のタスクプログラムで行う場合もある。タスクプログラムでデバイス処理を行うと、デバイス処理プログラムの変更が簡単である反面、デバイス応答速度が遅くなるという問題がある。マルチタスクオペレーティングシステム下でタスクプログラムが動作する時には、複

数のタスクプログラムの優先順位を判定し、どのタスクプログラムを止めて、どのタスクプログラムを動かすかという処理が必要である。この処理をスケジュール時間とか、ディスパッチ時間と呼ぶ。この時間がかかればかかるほど、該当デバイスが処理を要求しても、それに応答する時間が遅れる。デバイスが高速になればなるほど、この時間が問題となる。しかしシステムコールをもちいたデバイス処理部 5 を起動する方式は、この応答速度が優れている。なぜならばスケジュール時間無しに、該当デバイス処理を開始できるからである。

第 4 の実施例は、デバイス管理テーブル 6 の情報を再配置する機能を持つものである。端末器内には、高速処理が必要なデバイス 7 も存在する。優先的に処理されるべきデバイス 7 が、高速に検索できるようにデバイス管理テーブル 6 を構成しなければならない。しかし第 1 や第 2 の実施例では、当初に決めたデバイス情報の配列を変えられない。そこで高速処理が必要なデバイス 7 に対し

しておく必要もある。

第 5 の実施例として、第 6 図のように、デバイス処理部 5 を直接デバイス 7 のハードウェアの制御を行う機能を有するデバイスドライバ部 8 と、ハードウェアのみでは実現不可能な機能を拡張し、デバイス処理結果をメールボックス部 1 に通知するデバイスハンドラ部 9 で構成したものを説明する。

例えば、シリアル通信処理を行うシリアルポート (RS-232C 等) の制御用デバイス (インテル社 8251) は、ブロック転送等の高度な処理は持たない。ただバイト転送しかサポートしていない。そこである連続したデータ列を受信したい場合は、このデバイス 7 を指定回数連続して動作させる必要がある。このバイト転送をサポートするのがデバイスドライバ部 8 で、この指定回数になるまでデバイス 7 を動作させるのがデバイスハンドラ部 9 である。デバイス処理部 5 の作業を、デバイスドライバ部 8 と、デバイスハンドラ部 9 に分割した理由は次の通りである。ハードウ

て、検索順序が先になるようにデバイス管理テーブル 6 内の配置を入れ換えるものである。これは、高速デバイス 7 の情報を一番頭にもっていき、後は順に繰り下げていく方式をとってもよいし、先頭にあるものと追加した高速デバイス 7 を入れ換える方式をとっても良い。最初に登録してあるデバイス 7 が、高速処理を要求されている順に配置するように構成し、追加デバイス 7 を追加したい番号の所に挿入し、後は順に繰り下げる機能を持つても良い。この時に同時にデバイス処理部 5 内の各デバイス処理プログラムも再配置し、メモリの歯抜けを埋めるような、ガーベージコレクションの機能を実現できれば、なお好ましい。デバイス処理プログラムを並べ換える際、先にも述べたが、プログラムがすべてリロケータブルな形式になっている必要がある。これはどこにプログラムが配置されても動作を可能にするためである。またマイクロプロセッサによっては、インテル社 8086 のように「FAR」という命令を用いて、大規模なメモリ構成を意識したプログラムを開発

エが固定されると、そのデバイス 7 を用いて出来るハードウェア処理は固定される。しかし同じデバイス 7 を用いて、違った使い方をしたい時がある。例えば、1 バイト目はあるパターンの文字入力待ち、次にある文字列がくるまでデータを受信続け、最後に通信検査用のデータの入力を受け付ける通信規約例を考える。この内ハードウェア側で出来るのは、1 バイト入力の機能だけであるから、受信データのうちどこからを有効データと判定し、どこで終了し、どこで検査データが来るかを判別するかは、すべてソフトウェアで行う必要がある。しかしこの通信規約は、相手方の都合で変更される可能性もあるし、また同じデバイス 7 を使って異なる装置に接続され、全く異なった通信規約を実現しなければならないこともある。そこでハードウェアに依存し、ハードウェアが変わらないかぎり変更されないデバイスドライバ部 8 と、使用したい条件を取り入れ、適宜入れ換えて使用するデバイスハンドラ部 9 に分割したものである。もしくは、ハードウェアそのものが新し

いものに変更されたとしても、そのデバイスドライバ部8だけを変更すれば、あとの部分は変更しなくてもすむような、プログラムの階層構造実現のねらいもある。

タスク処理部3からは、デバイスドライバ部8がいつ動いているかはわからない。デバイスハンドラ部9から通知されるメールメッセージによって、その存在がタスク処理部3に知らされる仕組みとなる。タスク処理部3とデバイスドライバ部8とデバイスハンドラ部9は、独立に動作可能である。互いに連絡をとるための手段は、コマンド発行とメールメッセージ受信のみである。コマンド解析部2は、必要なコマンドを受け付けた後で、デバイス管理テーブル6を見に行き、初めてデバイスドライバ部8の存在を知る。またデバイスハンドラ部9は、デバイスドライバ部8が起動された後で、初めて動作を開始する。

第6の実施例は、一つのデバイスドライバ部8に対して複数のデバイスハンドラ部9を持つように構成したものである。第5の実施例内で述べた

ハンドラ部9は、仕様毎にそれぞれ独立に構成でき、小規模の仕様変更にも柔軟に対応できる。また機能が同じ複数のデバイスドライバ部8が存在する場合は、複数のデバイスドライバ部8と、同一のデバイスハンドラ部9を組み合わせても可能である。

ただしコマンド解析部2への依頼は、デバイスドライバ部8とデバイスハンドラ部9のペアの情報を送る必要がある。このペアの状態をデバイス管理テーブル6内に示したものが、第7図である。デバイスドライバ番号「01」に対し、デバイスハンドラ番号「21」と「22」が対応している。コマンド解析部2はデバイスドライバ「01」とデバイスハンドラ「21」を指定すると、獲得状態を「1」に変更する機能を持つことで（通常は「0」）、どのデバイスハンドラ部9が選択されたかがわかる仕組みとなっている。デバイスハンドラ部9及びデバイスドライバ部8は第1の実施例や第2の実施例と同様にそれぞれ追加、変更出来るような構成をとることが望ましい。必要に応

ように、一つのデバイス7に対して使用方法の異なる複数の要求を実現するために、デバイスハンドラ部9を複数持つようにし、コマンド解析部2が、デバイスドライバ部8とデバイスハンドラ部9を選択して起動をかける必要がある。例えば第8図に示すように一つのデバイス7に対し、一つのデバイスドライバ部8と、二つのデバイスハンドラ部91、92で構成し、必要に応じてどちらかのデバイスハンドラ部9を選択する。もしくは後から追加したデバイスハンドラ部93を選ぶ時もある。このように分割した構成をとることにより、次のような利点もある。従来は2種類の機能を1本のデバイス処理プログラムで実現したとすると、プログラム内でこのうちのどちらかを選ぶ選択枝が必要であり、また2種類のパラメータの両方に対応するプログラムを作成する必要があった。しかしデバイスハンドラ部9が予め選択されていれば、分類処理が不要になり、パラメータ処理も簡素化され、その結果、高速かつ柔軟な処理体系がとれるようになった。さらにデバイス

に単独でも追加、変更しシステムの柔軟さに追随できる構成とする。

第7の実施例を次に示す。第8図は、デバイス管理テーブル6をデバイステーブル64と、ハンドラテーブル65の2種類で構成し、それぞれの連結関係を動的に変化させることができるようにしたものである。デバイステーブル64内の、デバイスハンドラアドレスを示す情報を書き換える仕組みを持つことで、異なるハンドラテーブル65を指定することもできる。第8図内で63は、デバイステーブル64内で指定されるドライバプログラム、66はハンドラテーブル65内で指定されるハンドラプログラムを指す。ここでは、デバイスドライバ「01」が指定され、デバイスハンドラAのハンドラテーブル65からデバイスハンドラ「01」が選択されたことを示している。このように構成することで、デバイス管理テーブル6を、追加、変更デバイス7の数に応じて動的に作る事が出来る。第7図のように構成すると、デバイス管理テーブル6に予め空き領域を構成し

ておき、その部分に追加することになるが、第8図のように構成すると、デバイステーブル64の方にのみ空き領域を作っておけば良く、使用目的に応じた最適のハンドラ情報の追加に、柔軟に対応することが出来る。理由は、デバイステーブル64内のデバイスハンドラアドレスを書き換えれば、ハンドラテーブル65はメモリ空間のどこに作っても良いので、ハンドラテーブル65のサイズには左右されない構成となるからである。

第8の実施例として、コマンド解析部2がデバイス管理テーブル6内のデバイスステータスを見ることが、マルチタスクオペレーティングシステム下で、同時に同一デバイス7の競合使用を禁止するように構成した例を示す。第3図及び第7図、第8図の獲得状態とは、このデバイスステータスの1実現例である。該当するデバイスを指定、選択した後は、コマンド解析部2が、デバイスのステータスを獲得状態に設定する。このステータスはデバイス7の獲得もしくは使用状態を示す情報を持つ。すでにタスク処理部3内のあるタスクプ

ログラムがデバイス7の獲得要求を出し、使用を開始した後で、別のタスクプログラムが、同じデバイス7の獲得要求を出したとしても、ステータス情報をみて、内容が獲得状態であれば、あとの獲得要求はキャンセルされ、前のタスクプログラムの要求が完全に実現し終わるまで待たされる。このステータスは、デバイス処理部5もしくはデバイスハンドラ部9が処理を終了し、メールボックス部1へメッセージを通知する直前に解除する。ただし、メールボックス部へメッセージが届くまでは、次のデバイス獲得要求は受け付けないようにした方が良い。これは、メッセージを届ける前に、次のデバイス処理が開始され、次のメッセージが前のメッセージを消してしまったり、二つのメッセージが混じり合ったりするのを避けるためである。このステータスには、獲得、未獲得の2状態だけでなく、獲得状態であっても、現在待ち状態中であることを示す待機中や、現在処理の執行中ということを区別して格納することもできる。もしくはデバイス7に故障が見つかった時には故

障中であるとか、外部の接続機器の未接続等でデバイス7は登録されていても利用できないということを、使用者に知らせるための情報を格納するようにも構成できる。

本発明の効果をまとめると、第1に、デバイス7を動作させるデバイス処理部5の存在は、全てデバイス管理テーブル6に記述されており、コマンド解析部2はこの内容を検索するだけで、内部のデバイス情報が把握できること。第2にデバイス処理の終了はメールメッセージを持つだけの構成としたため、一つのデバイス7の処理の終了を待つことなく、複数のデバイス7に対し処理要求を出せるため、複数のタスクプログラムを起動せず、一つのタスクプログラムだけでもデバイス7の多重処理が可能になったこと。第3にデバイス7の変更、追加出来る構成を持つことで、システムの柔軟性が高まったこと。第4に特別な入出力コマンドを持たなくても、追加、変更されたデバイス7に対するコマンド解析が可能なこと。第5に高速処理が必要なデバイス7を優先的に配置す

ることで、更にデバイス7の高速処理が可能になること。第6にデバイスドライバ部8とデバイスハンドラ部9をもつことで、複数のデバイス処理要求に、高速かつ柔軟に対応できること。第7にこれらのデバイス7の競合状態を回避できる手段をもつことである。

発明の効果

以上述べてきたように、本発明の情報処理端末器はデバイスを制御するデバイス情報を全てデバイス管理テーブル6にもち、デバイス使用時に、コマンド解析部がこの内容を参照することで、該当デバイスの処理を行うことが可能で、かつこのデバイス情報を変更、追加が可能な構成を持つことで、デバイス構成要素の変更や、仕様の変更等に柔軟に対応できる情報処理端末器を提供することができる。

4. 図面の簡単な説明

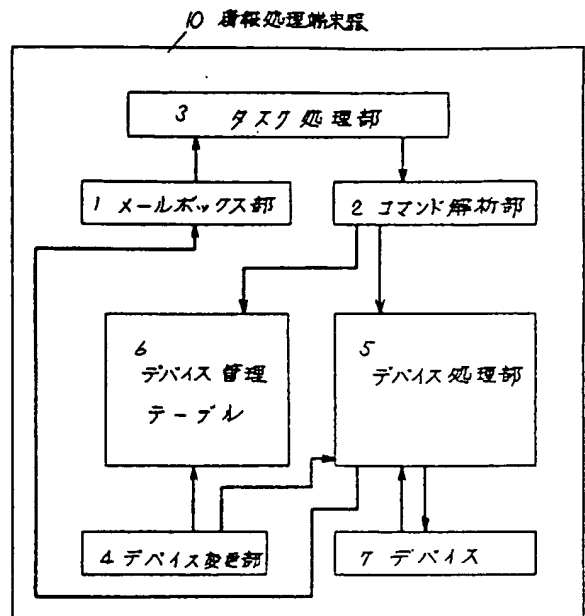
第1図は第1の実施例の情報処理端末器の構成図、第2図は第1図の情報処理端末器の処理手順を示す図、第3図はデバイス管理テーブルの構成

図、第4図はデバイス管理テーブルのメモリ構成図、第5図は第5の実施例の情報処理端末器の構成図、第6図は第6の実施例のデバイスドライバ部と、デバイスハンドラ部の構成図、第7図は第6の実施例のデバイス管理テーブルの構成図、第8図は第7の実施例のデバイス管理テーブルの内容を示した情報関連を示す図、第9図は従来の情報処理端末器の構成図、第10図は従来の情報処理端末器の処理手順を示す図である。

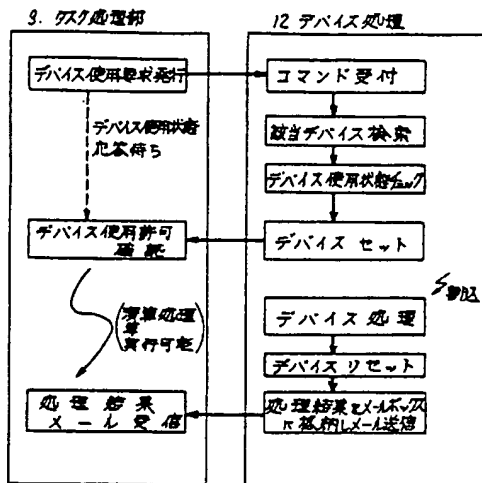
1……メールボックス部、2……コマンド解析部、3……タスク処理部、4……デバイス変更部、5……デバイス処理部、6……デバイス管理テーブル、7……デバイス、10……情報処理端末器。

代理人の氏名 弁理士 栗野重孝 ほか1名

第 1 図



第 2 図



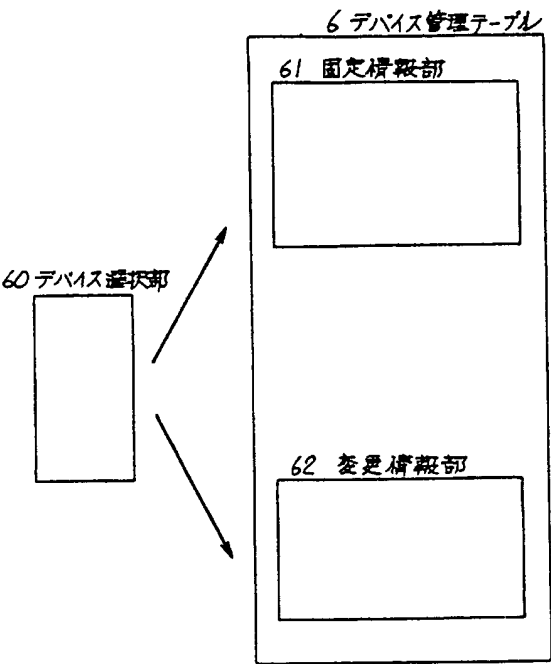
第 3 図

6 デバイス管理テーブル

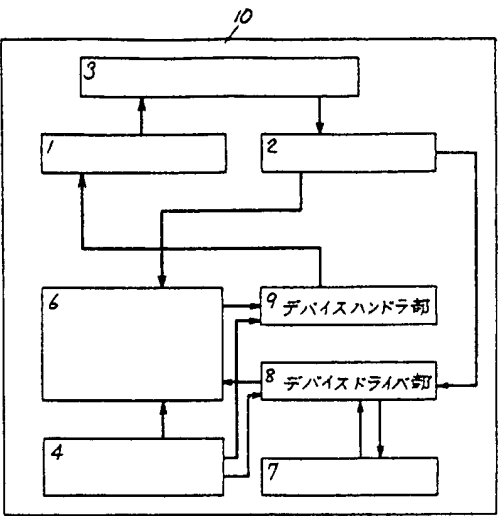
デバイス名	デバイスプログラムアドレス	獲得状態
A	F0000H	1
B	F1000H	1
C	F1800H	0
...
K	F2600H	0
C	F4000H	0

変更領域

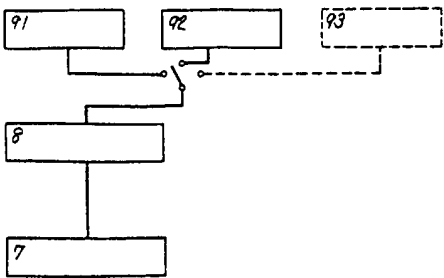
第 4 図



第 5 図



第 6 図

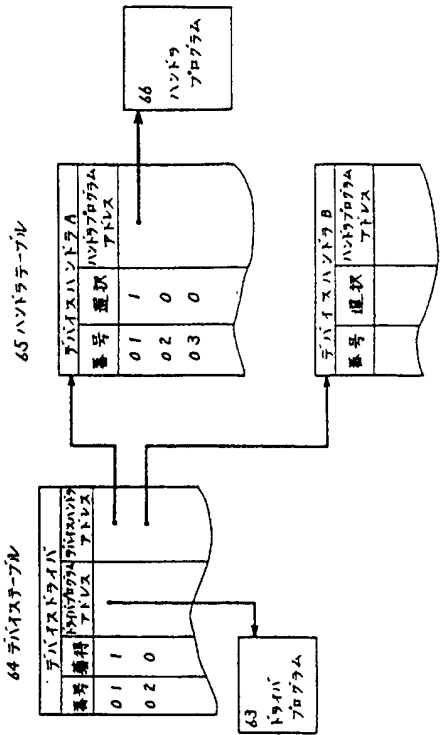


第 7 図

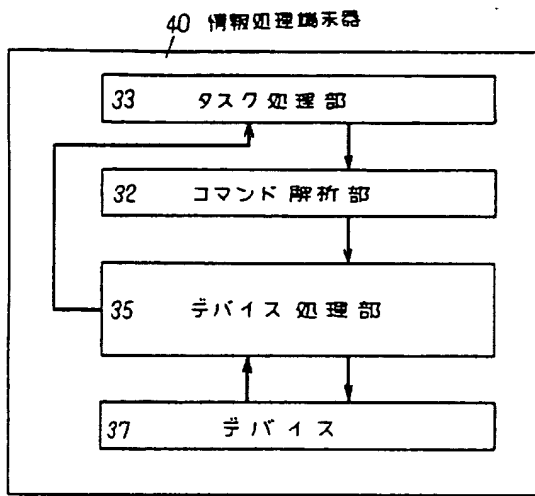
6 デバイス管理テーブル

デバイスドライバ		デバイスハンドラ		備 考
番号	アドレス	番号	アドレス	
01	F2000h	21	F4200h	1
01	F2000h	23	F4600h	0
02	F2340h	25	F5400h	1
03	F3000h	29	F5800h	0
...	
08	F3F00h	52	FE000h	0

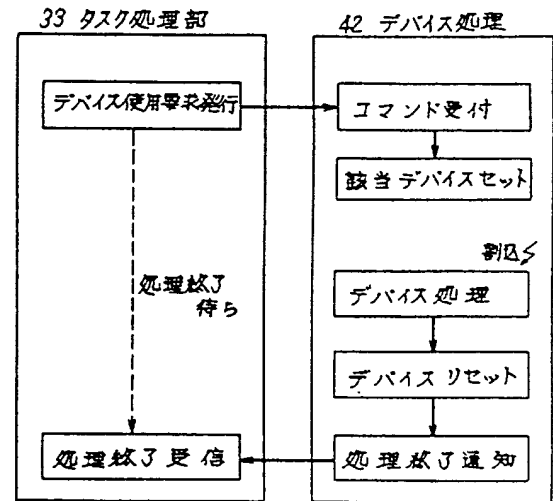
第 8 図



第 9 図



第 10 図



THIS PAGE BLANK (USPTO)